REMARKS

In response to the final Office Action mailed on July 17, 2007, Applicant respectfully requests reconsideration. Claims 1-12 were previously pending in this application. By this amendment, claims 1, 6 and 9 are amended. No claims are added or canceled herein. As a result, claims 1-12 are pending for examination with claims 1, 3, 6, 7, 9 and 11 being independent.

Applicant's representatives appreciate the courtesy extended by Examiner Johnson in agreeing to the telephone conference conducted on September 14, 2007, the substance of which is summarized in the arguments provided below.

I. Allowable Subject Matter

Applicant appreciates the Examiner's indication that claims 3, 4, 7, 8, 11 and 12 are allowed.

II. Rejections Under 35 U.S.C. § 103

The Office Action rejects claims 1, 2, 5, 6, 9 and 10 as purportedly being unpatentable over U.S. Publication No. 2002/0078333 (Inoue) in view of Applicant's Background. While Applicant believes the claims, as previously presented, distinguish over the alleged combination, Applicant has amended the rejected claims to clearly distinguish over the alleged combination.

The final Office Action asserts that Inoue discloses in paragraph **[0011]** "[p]roviding the loop top instruction address to an instruction fetch stage of the pipelined processor if it is determined that the next instruction is the loop bottom instruction, and *fetching the loop top instruction prior to completing the decoding of the loop bottom instruction*" (Page 3 of the Office Action). (Emphasis added). As discussed during the telephone conference, Applicant agrees that Inoue must detect the loop bottom instruction in some manner. However, Inoue says nothing about how this is done, or more importantly, when this is done. This is true for the following reasons: 1) Inoue is directed to setting up a hardware loop and says very little about the loop operation after setup; and 2) as the final Office Action concedes, Inoue does not disclose variable width instructions. As discussed during the telephone conference, constant width instruction pipelines do not have the same problems associated with loop bottom detection as variable width instruction

pipelines because the width of each instruction in a constant width instruction pipeline is known *a priori* (i.e., all instructions have the same instruction width). Therefore, the pipeline does not have to decode an instruction to obtain the instruction width for comparison with the loop bottom offset because the instruction width is already known. For example, a comparison to check whether the program counter (PC) plus the instruction width is equal to the loop bottom address can be performed without decoding the instruction because the instruction width is always known in constant width instruction pipelines. This comparison, however, requires decoding the instruction to obtain the instruction width in variable instruction width pipelines.

The final Office Action combines Applicant's Background with Inoue to allegedly cure the deficiency in Inoue (i.e., that Inoue does not disclose variable width instruction pipelines). In particular, the Office Action asserts that Applicant's specification discloses the use of variable-width instructions and that it "would have been obvious at the time of the invention for one of ordinary skill in the art to allow the invention of Inoue to incorporate variable length instructions" (Page 4 of the Office Action). While Applicant does not agree that one of ordinary skill in the art would have been motivated to combine Inoue (i.e., a constant width instruction pipeline) with the variable width instruction pipelines described in Applicant's Background, the alleged combination still does not disclose each of the limitations in Applicant's claims.

In particular, while Applicant agrees that the Background discloses the use of variable width instructions and that processors for processing variable width instructions were known, the Background describes prior art processors to point out a shortcoming in conventional variable width instruction pipelines. Specifically, Applicant reports in the Background that because it is not possible to know the loop end condition in variable width instruction pipelines in time to slot the next fetch in the pipeline, the processor will stall (Page 3, lines 8 and 9 in Applicant's Background). That is, because the width of an instruction is not known until the instruction is decoded, loop bottom detection cannot be made prior to fetching the next instruction. As a result, the processor will fetch the instruction at the next address in memory, rather than correctly fetching the loop top instruction (which resides at a memory location before the loop bottom instruction, not after). As a result, by the time the loop bottom instruction is detected (i.e., after it is decoded), incorrect instructions have already been inserted into the pipeline, which causes the pipeline to stall while the

loop top instruction is fetched and the incorrect instruction(s) purged.

In the Background, Applicant also describes prior art processors that provide a loop buffer to cache a few instructions at the top of the loop that can be inserted into the pipeline when the loop bottom instruction is detected after decode, a solution which is also vulnerable to pipeline stalls (Page 3, lines 10-21 of the Background). Applicant's Background nowhere discloses a variable width instruction pipeline capable of detecting a loop bottom instruction prior to the instruction being decoded.

Accordingly, a combination of Inoue with Applicant's Background, at best, results in a variable width instruction pipeline that detects loop bottom instructions during the decode stage, and thus is vulnerable to incorrect instruction fetches that either cause a stall or triggers a "loop buffer" to inject instructions into the pipeline. In either case, the combination does not result in a variable width instruction pipeline configured to detect a loop bottom instruction prior to decoding the instruction. Inoue describes a constant width instruction pipeline but is completely silent with respect to variable width instructions or any of the problems associated therewith. Applicant's Background describes variable width instruction pipelines that are incapable of detecting the loop bottom instruction prior to decoding the instruction.

Therefore, the only way to arrive at a variable width instruction pipeline capable of detecting a loop bottom instruction before the instruction is decoded is by using Applicant's own insight as described in the Summary and Detailed Description of Applicant's specification. Applicant alone has recognized that the information in the loop setup instruction coupled with information obtained from decoding the instruction before the loop bottom instruction may be used to detect the loop bottom instruction prior to actually decoding the instruction (Page 8, line 23 – page 10, lines 25). Thus, the alleged combination of Inoue and Applicant's Background fails to disclose loop bottom detection in a variable width instruction pipeline prior to decoding the loop bottom instruction. Accordingly, the claims patentably distinguish over the alleged combination, as discussed in further detail below in connection with each of the claims.

A. Claims 1, 2 and 5

Claim 1, as amended, recites a method for processing variable width instructions in a pipelined processor, comprising decoding instructions to identify a loop setup instruction having a

loop setup instruction address to determine a loop top offset indicative of a loop top instruction address of a loop top instruction and a loop bottom offset indicative of a loop bottom instruction address of a loop bottom instruction, **decoding variable width instructions following the loop setup instruction to detect the loop bottom instruction, a width of each of the variable width instructions being determined during decoding**, decoding a current instruction of the variable width instructions, the current instruction having a current instruction address and a current instruction width, fetching a next instruction of the variable width instructions, **determining if the next instruction to be decoded is the loop bottom instruction based, at least in part, on the current instruction address, the current instruction width, the loop setup instruction address and the loop bottom offset, the determination being made prior to decoding the next instruction**, and providing the loop top instruction address to an instruction fetch stage of the pipelined processor if it is determined that the next instruction is the loop bottom instruction.

First, claim 1 requires decoding variable width instructions and that the width of the variable width instructions are determined during decoding. Next, claim 1 requires determining if the next instruction to be decoded is the loop bottom instruction based on information about a current instruction being decoded and information obtained from the loop setup instruction. Finally, claim 1 requires that this determination be made prior to decoding the next instruction. The alleged combination of Inoue and Applicant's Background nowhere discloses or suggests decoding variable width instructions and determining if a next instruction to be decoded is a loop bottom instruction wherein "the determination [is] made prior to decoding the next instruction," as recited in claim 1. Therefore, claim 1 patentably distinguishes over the alleged combination and is in allowable condition. Claims 2 and 5 depend from claim 1 and are allowable based at least upon their dependency.

B. Claim 6

Claim 6, as amended, recites **a apparatus for processing variable width instructions in a pipeline processor**, comprising **an instruction decoder** configured to decode a loop setup instruction, having a loop setup instruction address, to obtain a loop top offset indicative of a loop top instruction address of a loop top instruction and a loop bottom offset indicative of a loop bottom instruction address of a loop bottom instruction and **configured to decode variable width**

instructions following the loop setup instruction to determine an instruction width of each of the variable width instructions, an instruction fetch stage configured to fetch instructions to be decoded by the instruction decoder, registers for holding the loop setup instruction address and the loop bottom offset, respectively, and **a loop bottom detector configured to determine if a next instruction to be decoded and currently in the instruction fetch stage is the loop bottom instruction prior to decoding the next instruction, the determination being made based, at least in part, on a current instruction address and current instruction width of a current instruction being decoded by the instruction decoder**, the loop bottom detector configured to provide the loop top instruction to the instruction fetch stage if the loop bottom detector determines the next instruction is the loop bottom instruction.

Nowhere does the alleged combination of the Inoue and Applicant's Background disclose or suggest an apparatus for processing variable width instructions in a pipeline processor including an instruction decoder to decode variable width instructions to determine the instruction widths and "a loop bottom detector configured "to determine if a next instruction to be decoded and currently in the instruction fetch stage is the loop bottom instruction prior to decoding the next instruction," as recited in claim 6. Therefore claim 6 patentably distinguishes over the alleged combination and is in allowable condition.

C. Claims 9 and 10

Claim 8, as amended, recites **an apparatus for processing variable width instructions in a pipelined processor**, comprising **means for decoding** a loop setup instruction, having a loop setup instruction address, to obtain a loop top offset indicative of a loop top instruction address of a loop top instruction and a loop bottom offset indicative of a loop bottom instruction address of a loop bottom instruction and for decoding **variable width instructions following the loop setup instruction to determine an instruction width of each of the variable width instructions**, means for fetching instructions to be decoded by the means for decoding, means for holding the loop setup instruction address and the loop bottom offset, and **means for determining if a next instruction to be decoded is the loop bottom instruction prior to decoding the next instruction, the determination based, at least in part, on a current instruction address and current instruction width of a current instruction being decoded by the means for decoding**, and configured to

provide the loop top instruction to the means for fetching instructions if the next instruction is determined to be the loop bottom instruction.

Nowhere does the alleged combination of the Inoue and Applicant's Background disclose or suggest an apparatus for processing variable width instructions in a pipeline processor including means for decoding variable width instructions following the loop setup instruction to determine an instruction width of the variable width instructions and "means for determining if a next instruction to be decoded is the loop bottom instruction prior to decoding the next instruction," as recited in claim 9. Therefore claim 9 patentably distinguishes over the alleged combination and is in allowable condition. Claim 10 depends from claim 9 and is allowable based at least upon its dependency.

CONCLUSION

In view of the above remarks, Applicant believes the pending application is in condition for allowance, and a Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully submitted,
Christopher M. Mayer, Applicant

By: William R. McClellan

William R. McClellan, Reg. No. 29,409
Wolf, Greenfield & Sacks, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210-2211
Telephone: (617) 720-3500

Docket No. A0312.70524US00
Date: September 17, 2007
x09/17/07x